Physica A 473 (2017) 89–96

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Evolutionary algorithm and modularity for detecting communities in networks

Saoud Bilal [a,*], Moussaoui Abdelouahab [b]

[a] Department of Computer Science, Faculty of Exact Sciences, University of Bejaia, 06000, Bejaia, Algeria
[b] Intelligent Systems Laboratory (LSI), Faculty of Sciences, University of Setif1, 19000, Setif, Algeria

HIGHLIGHTS

- In this paper, we introduce our method for community detection.
- Our method designed to detect community structure for unweighted and undirected networks.
- We used an evolutionary algorithm to find the first community structure.
- We used the modularity in the merging process to find the final community structure.
- Finally we test our method on both artificial and real networks.

ARTICLE INFO

Article history:
Received 11 July 2016
Received in revised form 7 December 2016
Available online 4 January 2017

Keywords:
Community detection
Networks
Evolutionary algorithms
Normalized mutual information
Modularity

ABSTRACT

Evolutionary algorithms are very used today to resolve problems in many fields. There are few community detection methods in networks based on evolutionary algorithms. In our paper, we develop a new approach of community detection in networks based on evolutionary algorithm. In this approach we use an evolutionary algorithm to find the first community structure that maximizes the modularity. After that we improve the community structure through merging communities to find the final community structure that has the high value of modularity. We provide a general framework for implementing our approach. Compared with the state of art algorithms, simulation results on computer-generated and real world networks reflect the effectiveness of our approach.

1. Introduction

Networks can represent many systems shush as biology, computer science, linguistics, etc. A network has a set of vertex (or node) and a set of edge. The vertexes represent the individuals members of the system and the edges represent the links between nodes in a defined relationship of the system. For example, in a social network, a person is represented by a node, and the social interaction of relationship between two people is represented by an edge. When a system is modeled by a network, it helps to understand the system easily. Lots of studies have done around networks and how to use them.

Networks have some common features. The most known feature, that helps to understand the network structure, is the existence of parts more densely connected than other parts. These parts, which are set of nodes and edges, are called communities. So, the community can be defined as a group of nodes much more strongly connected to each other than other nodes. In addition, there are many definitions of the community.

* Corresponding author.
E-mail address: bilal340@gmail.com (S. Bilal).

http://dx.doi.org/10.1016/j.physa.2017.01.018
0378-4371/

The detection of communities in networks has become a new field of research. The community detection is an NP-hard problem. Many methods have been proposed to find the best community structure in network, although the problem is not yet satisfactorily resolved. These methods can be classified according to the type of networks (unipartite or bipartite, weighted or unweighted) and the community structure (disjoint or overlapping) [1,2]. In this paper we focus on methods for unipartite, unweighted networks to detect disjoint community structure.

Several community structure detection methods have been proposed in different areas (social networks, metabolic networks, communication networks, etc.). Among the most important methods we find, Kernighan–Lin method [3], which uses a bisection algorithm to find the graph cut which minimizes the number of edges between two groups. The Girvan and Newman method [4] is based on the betweenness centrality (number of shortest paths passing through as edge). Radicchi et al. method [5] is based on clustering coefficient of edges. The method removes the edge of lower coefficient at each step. Furtunato et al. method [6] is a variant of Girvan and Newman method, based on the information centrality. Clauset et al. method [7], an improved Newman method [8], is based on the greedy optimization of modularity. The method proposed by Donetti and Muoz in [9] uses a hierarchical clustering based approach with eigenvectors of the Laplacian matrix of the graph to find the similarity between nodes. Pons and Latapy [10] in their work proposed a hierarchical clustering method measuring the similarity between nodes in graph based on random walks. The method proposed by Rosvall and Bergstrom [11], also uses the concept of random walks and entropy communities to find the community structure. Blondel et al. method [12] is a heuristic method that is based on modularity optimization. It starts from each node as a community, and merged communities based on the modularity criterion. This operation is repeated several times on the set of nodes until no further optimization is possible. Raghvan et al. method [13] is based on label propagation. Initially, every node in the graph is initialized with a unique label and at every step of the method each node takes the label that most of its neighbors currently have. The iterative process converges when labels cannot be changed. The method proposed by Raghvan et al., is non-deterministic method. Xiang et al. method [14] proposed a new metric to quantify the structural similarity between sub-graphs, based on this subgraph similarity an algorithm for community detection is designed. Among the methods that we can find in the literature based on the genetic algorithm we can list the method of Ronghua et al. [15–17] which is based on the optimization of the modularity function through a genetic algorithm. And finally we cite some methods based on the similarity between nodes, optimization of modularity, cliques in the graph etc. [18–21].

In this paper, we propose a new method for unweighted and undirected networks to find their community structure. Our method is based on the similarity between nodes in term of shared neighbors and the optimization of the function of quality called modularity. For the optimization of the modularity we used the evolutionary algorithms. In other words, for a graph $G(V, E)$ our method can find the most optimal community structure $\pi = \{c_1, \dots, c_k\}$, such as $\bigcup_{i=1}^{k} = V$ and $c_i \neq \emptyset$, $c_i \cap c_j = \emptyset$ (for $i, j = 1 : k$). The paper is organized as follow. The concept of evolutionary algorithm is presented in Section 2. Our approach is detailed in Section 3. Experimental results are shown in Section 4. Finally, we conclude in Section 5.

2. Evolutionary algorithms

The evolutionary algorithms (*EA*) [22] are heuristic that mimics biological evolution as a problem solving strategy. Given a specific problem to solve, the input to algorithm is a random population of solutions. Each possible solution, which is member of the population, is called individual. These individuals are encoded in some way (binary for example). To evaluate an individual there is a metric called a Fitness function. In addition, to generate or create new individuals, which will replace some individuals of the population, some genetic operators are used in each step of the algorithm such as Mutation, Crossover, etc.

In EA process some individuals are kept and allowed to reproduce. Multiple new individuals are generated based of them; the generation process is guided by the genetic operations (selection, mutation and crossover). These offspring form a new set of individual solutions (new generation), and they are subjected to a second step of fitness evaluation. Those individual solutions which were worsened are again deleted; purely by chance, the random variations introduced into the population may have improved some individuals, making them into better, more complete or more efficient solutions to the problem. Again these successful individuals are selected and copied over into the next generation with random changes, and the process repeats. The expectation is that the average fitness of the population will increase each step, and so by repeating this process many times. Finally, the best solutions to the problem can be discovered. Fig. 1 illustrates the evolutionary algorithms principle.

2.1. Representation

Individuals should be represented by code to permit the EA works on them. The classic EA choice is to encode each individual as binary. However, any representation is permitted for which suitable variation operators can be devised.

2.2. Selection

This operator may be the most important because it allows keeping or deleting an individual in a population. As a general rule, the probability of keeping an individual is directly related to its fitness evaluation value. There are several methods